AEC Computing Facility

AEC RESEARCH AND DEVELOPMENT REPORT

PHYSICS NYO-7692

THE NYU OMNIFAX COMPILER AND LIBRARY OF SUBROUTINES

Florence F. Ragusa

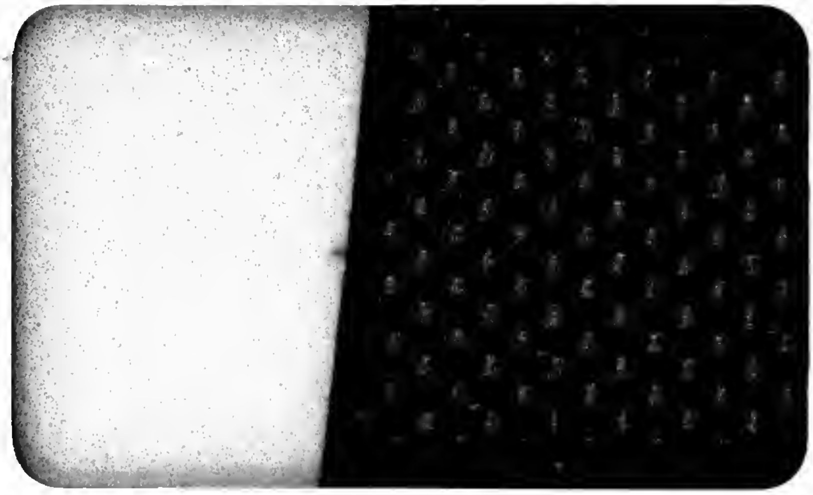
Sandra Zucker

October 15, 1956

institute of mathematical sciences

NEW YORK UNIVERSITY

NEW YORK, NEW YORK



This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission to the extent that such employee or contractor prepares, handles or distributes, or provides access to, any information pursuant to his employment or contract with the Commission.

AEC Computing Facility
Institute of Mathematical Sciences
New York University

PHYSICS NYO-7692

THE NYU OMNIFAX COMPILER AND LIBRARY OF SUBROUTINES

Florence F. Ragusa
Sandra Zucker

October 15, 1956

CONTRACT No. AT(30-1)-1480
with the United States Atomic Energy Commission

Table of Contents

page

Abstract..8

Part I Omnifax Compiler...............................9

Section

1. General Description and Procedure..................9

2. Subroutine Library.................................10

   A. Subroutines.....................................10

   B. Subroutine words................................11

   C. Library of Subroutines..........................13

3. Directory..14

   A. Description.....................................14

   B. Procedure.......................................15

   C. Flexibility.....................................15

   D. Types of Directory Items........................16

4. Modifications (Mod I and II).......................18

   A. Description.....................................18

   B. Parameters......................................18

   C. Keys..19

5. Parallel Edit......................................21

6. Method of Operation................................21

   A. Control word....................................21

   B. Servos..22

   C. Breakpoints.....................................23

D. Type-outs 23

E. Restarts 25

7. Example 25

Directory 25

Subroutine 26

Part II Index - NYU Library of Subroutines 27

Arithmetic Routines 27

Interpretive Floating Decimal Routine (AF)....27

Composed Floating Decimal ....(CD)....27

Floating Point Arithmetic ....(FP)....27

Floating Decimal Routine ....(FR)....27

Complex Number Arithmetic (floating)..(CE)....27

Complex Number Arithmetic (fixed).....(CX)....28

Debugging Routines 28

Generator for Debugging Routines......(GR)....28

Locator...............................(LC)....28

Address Search........................(AS)....28

Automatic Dumping Routine.............(DB)....28

Memory Dump Comparer..................(DC)....28

Storage Dump Routines..............(DP,DS)....29

Q,T,U, Tracer (Typewriter)............(TR)....29

Q,T,U Tracer (Tape)...................(TE)....29

Tracer Counter..........................(TC)...29

Register A Monitor (Typewriter)...........(MN)...29

Register A Monitor (Tape)................(ME)...29

Follower I..............................(FL)...29

Differential Equation Routines.........................30

Differential Equation Solution............(DF)...30
(floating)

Differential Equation Solution............(DY)...30
(fixed)

Differential Equation Solution............(DM)...30
(Milne)

Edit and Output Routines...............................30

Sign Edit...............................(EA)...30

Edit Composed Floating Decimal Number.....(EB)...30

Edit Decomposed floating Decimal Number...(EP)...30

Edit Fixed Point Number..................(ES)...31

Carriage Return Counter..................(R̸C)...31

Packing Routine..........................(PA)...31

Block Output.............................(BL)...31

Block Output [V-Register]................(BV)...31

Exponentials and Roots.................................32

A^x...................................(AX)...32

10^x (floating)..............................(LH)..32

10^x (fixed)..................................(LS)...32

10^{-x} (fast-fixed)..........................(LV)...32

e^x (floating-routine FR)....................(LP)...32

e^x (floating-routine FP)....................(LQ)...32

e^x (fixed)...................................(LT)...32

Square Root (floating, composed, FR)......(NA)...33

Square Root (floating, decomposed, FR)....(NJ)...33

Square Root (floating-routine FP).........(NK)...33

Square Root (fixed)........................(NS)...33

$A^{1/p}$ (fixed)..............................(NX).. 33

Ignore Squeeze Routines...................................33

Ignore Squeeze Block Output I.............(GN)...33

Ignore Squeeze Block Output II............(IG)...33

Integration Routines......................................33

Simpson's Rule............................(QN)...33

Gauss' Method (floating).................(GM)...34

Gauss' Method (fixed)....................(GS)...34

Interpolation Routines.................................. 34

Aitken Interpolation (floating)...........(IP)...34

Lagrange Interpolation (floating).........(IL)..34

Logarithm Routines 34

Natural Logarithm (floating-routine FR)....(LF)...35

Natural Logarithm (floating-routine FP) ....(LG) ...35

Natural Logarithm (fixed)..................(LM) ...35

Logarithm to Base 10 (floating) ....(LN) ...35

Logarithm to Base 10 (fixed) ....(LU) ..35

Logarithm to Base 2 (floating) ....(LJ) ..35

Logarithm to Base 2 (fixed) ....(LX) ..35

Matrices and Linear Equations 35

Multiplication of Matrices (floating).....(MC) ...35

Number Theory 35

Factorization ....(FZ) ...35

Polynomial Routines 36

Polynomial Evaluation ....(PS) ...36

Sorting Routines 36

Sorting Routine I ....(ST) ...36

Trigonometric Routines 36

Sine-Cosine (composed floating)............(TD) ...36

Sine-Cosine (decomposed floating)..........(TK) ...36

Sine-Cosine (routine FP) ....(TL)....36

Sine-Cosine (fixed) ....(TV) ...36

Cosine πx (fast-fixed) ....(TU) ...36

Arctangent (continued fraction) ....(TY) ...37

Arctangent (polynomial) ....(TZ) ...37

Abstract

The New York University Compiler System previously described in NYO-6478 has been revised to become an integral part of the single general NYU service routine called Omnifax (NYO-6485). This report describes all aspects of the use of the Omnifax Compiler.

A complete index of the NYU Library of Subroutines currently available for use with the Compiler is included in the report. The listing supersedes that given in NYO-6483.

Section 1. General Description and Procedure

Description

The Omnifax Compiler is a program which takes as input a number of isolated sections of coding (called subroutines) together with instructions of how they are to be combined (the directory), and modifies and assembles them into a working program. The location of the subroutines in the memory for the running program is determined by the programmer.

Procedure

Subroutines already available in a general library, as well as new subroutines unique to a problem are utilized by the compiler to create the final program. A directory (described on page 14) is prepared by the programmer to specify which subroutines are to be used and how the final tape and the memory shall be organized.

Result

The addresses relative to each subroutine, and the cross references between subroutines are translated into absolute addresses in conformity with the specifications of the directory, and a final program tape is evolved. The Compiler permits both the program tape and the memory to be organized in whatever manner is best suited to the problem.

Parallel Edit

The Omnifax Compiler is able to put out on a specified tape information to be edited. The parallel edit, using this tape as input, produces a format giving the relative addresses and coding of the modified subroutines in their original form side by side with the absolute addresses and coding of the final program. The programmer is then able to see what was put into the compiler and what was the result of the compilation.

Section 2. Subroutine Library

A. Subroutines

Subroutines are the ingredients which the compiler assembles into a program. There are no restrictions on the length of any subroutine. Each subroutine is coded in relative form such that the first line of the routine is assigned the memory location 000 and succeeding lines are numbered consecutively.

1. Label line

Line 000 of the subroutine is reserved for the label line which carries the label of the routine, the number of lines of parameters in the directory for mod I or mod II (see page 18), the number of compilable lines in the subroutine, and a sentinel which identifies the line as the first line of a subroutine.

```
Example
    000
ABO MMM NNN 00i̸
```

where

AB - two Univac-character label assigned to the routine

MMM - number of lines of parameters in the directory for mod I or II

NNN - number of lines of the subroutine which will be transcribed onto the program tape

i̸ - symbol identifying this word as the label line of a subroutine

2. Subroutine Sentinel

The ending sentinel for a subroutine is a word of printer breakpoints, B̸B̸B̸B̸B̸B̸B̸B̸B̸B̸B̸B̸. It appears in the line which corresponds to the number NNN in the label line of the routine.

Note 1. Each subroutine must start a new block.

Note 2. The first line of a block on a subroutine tape should not end with an i̸ (ignore) unless that line is a label line.

B. Subroutine Words

The unit element on which the compiler operates is the individual line of coding within a subroutine. A line of coding consists of the customary left-hand and right-hand instructions to Univac, coupled with an additional character or characters imbedded in the word indicating how the addresses on that line are to be modified to make the subroutine capable of functioning correctly anywhere in the memory.

References:

The address portion of an instruction in a subroutine can have one of several meanings. It can be irrelevent (e.g. a "K" instruction), it can refer to a fixed part of the memory (absolute address), it can refer to a line in the subroutine itself or to a line in another subroutine.

The compiler is mainly concerned with the transformation of addresses in the coding of a subroutine from relative to absolute values. A relative address identifies the position of a line of coding with respect to the starting line of its routine. If an address refers to a line in the same subroutine, it is called an internal reference. If an address refers to a line in an entirely different subroutine it is called an external reference. A word may have two instructions, one having an internal reference and the other an external reference. This is called a mixed reference.

For external and mixed references the 2nd and 3rd digits carry the label of the other subroutine to which reference is made. The ninth digit in a line of coding is used to indicate to the compiler how that line should be modified. The ninth digit will be one of the following,

For Internal References

L	left-hand address is internal
R	right-hand address is internal
W	whole word is internal

For External References

X	whole line is external

For Mixed References

G	left-hand address is external; right-hand address is internal
D	right-hand address is external; left-hand address is internal

Note 1. References can be made to an external routine and an absolute address by coding the absolute address in the fourth zone alphabetic equivalents of the first zone numerics.

Note 2. It is not possible to refer relatively to two different external subroutines in the same line of coding.

An example of subroutine coding is given on page 26.

C. Library of Subroutines

Subroutines are stored on a tape which is called a library tape. The compiler can take subroutines from any one of three tapes.

General Library of Subroutines Tape

This tape contains a standard set of subroutines. The compiler assumes that these subroutines are stored alphabetically by label lines. The general library of subroutines is always mounted on servo 6.

Special Library of Subroutines Tapes (2)

This tape (or two tapes) contains subroutines which are unique to a problem. The compiler assumes that the subroutines are stored in the order in which they are listed in the directory, although this order is not required. Therefore it searches the library tape in a forward direction until an ending sentinel (see below) is reached. The tape is then searched in a backward direction.

The special library of subroutines tapes are either mounted on servo 3 or on the servo indicated in the input digit (digit 4) of the control word. (See page 21).

Library Sentinel

There must be an initial sentinel block whose first word is ZZZZZZZZZZZƵ and a final sentinel block whose first word is -ZZZZZZZZZZƵ on all of the library of subroutine tapes. The contents of the remainder of these two blocks are of no significance.

Section 3. Directory

A. Description

The directory is the control section for the compiler. It tells the compiler which subroutine to take from the library tape, on which library tape to look, and how to modify the routine once it is found.

The order in which subroutines are placed on the compiled program tape is determined by the order in which they

ERRATA

N.Y.U. Omnifax Compiler and Library of Subroutines - NYO-7692.

Please note that on page 14, under the heading Library Sentinel, the first word of the initial sentinel block should read **-ZZZZZZZZZ1̸** and the first word of the final sentinel block should read ZZZZZZZZZZ1̸ .

are listed in the directory. There are several other directory functions which are recognized by the compiler, and which help to make it a flexible system.

B. Procedure

The directory may be either prepared on tape, or typed directly into the memory at the time of compilation. It is limited in length to 3 blocks. The last item in the directory must be ZZZZZZZZZZZZ. The control word used to call the compiler into the memory tells Omnifax if the directory is to be typed in or read off tape. If the 6th digit of the control word is a

"O" (zero) then the directory is to be typed in in response to TYPE DIR;

"D" then the directory is read in off the first block of the input (I) tape (servo number is specified in the 4th digit of the control word);

"N" then the directory is already in the memory and the compiler starts processing the subroutines. (This is used when one has cleared "C" to restart.)

In making up the directory the programmer determines for himself how he wishes to utilize the memory for the problem. The processing of compiler items establishes the absolute starting address of each subroutine in turn. Relative references within the subroutine will then be adjusted by the compiler according to the starting line numbers listed in the directory.

C. Flexibility

The only consistency check applied by the compiler to the directory is a mod 10 check. A counter is used to keep track of the length of each routine. Before compiling each subroutine the least significant digit of this counter is checked against the least significant digit of the starting line number specified in the directory for the next item. Since no other check is used on the compiler the programmer may overlap several memory loads and may employ many of the tricks of direct programming.

Care must be exercised in assigning starting line numbers in the directory and insuring that read-in orders place subroutines into the memory locations for which the addresses have been adjusted.

D. Types of Directory Items

```
1.  ABO 000 NNN LLL (Subroutine specification)
      Z MMM
      I
```

D_1- D_2 AB Label line of subroutine to be compiled

D_3 0 Subroutine AB is stored in the library on servo 6 (General library)

Z Subroutine AB is stored on library on servo 3 (Special library)

I Subroutine AB is stored on library on servo I, 4th digit of control word (Special Library)

D_4-D_6 000 Compile

MMM Do mod I or II before compilation (See page 18) MMM are the number of parameters following this item

D_7- D_9 NNN number of compilable lines in subroutine AB

D_{10}- D_{12}	LLL	Starting line number of routine AB. Each address in the program which refers to subroutine AB will be modified by an increment of LLL.

2. STO S00 000 LLL (Reference specification)

D_1- D_2	ST	The reference label of a routine (not to be compiled).
D_3	0	Not used
D_4	S	Do not compile this routine
D_5- D_9	00000	Not used
D_{10}- D_{12}	LLL	External references to ST are modified by an increment of LLL.

3. 000 FFF 000 000 (Block fill specification)

D_1- D_3 , D_7- D_{12}	(0)	Not used.
	FFF	Fill remainer of block currently being written on program tape with skips. Repeated items of this type cause additional blocks of skips to be written.

4. 000 FF0 000 000 (Modulo 10 line fill specification)

D_1- D_3, D_6- D_{12}	0's	Not used.
D_4- D_5	FF	Fill remainder of current 10 word subblock of program tape with skips. Next subroutine starts in

a zero (mod 10) line. Repeated items of this type cause subblocks of skips to be written.

5. 000 FNN 000 000 (Arbitrary line fill specification)

$D_1 - D_3$, $D_7 - D_{12}$	0's	Not used.
$D_4 - D_6$	F(NN)	Fill the next NN location with skips, $0 \leq NN \leq 99$

6. ZZZ ZZZ ZZZ ZZZ (Directory end specification)
Terminal item of the directory.
Fill the remainder of the last block with skips.

A sample directory is given on page 25.

Section 4. Modifications (Mod I and II)

A. Description

The function of modifications in the compiler is to permit the transfer of half or full words of information (parameters) from the directory into preselected lines of the subroutines. Generalized subroutines can be altered by the compiler under directory control to particularize them for the problem being compiled. The lines of the subroutine can be filled by the compiler either by adding, (Mod I), or extracting (Mod II), half or whole word parameters from the directory.

B. Parameters

The directory supplies the parameters to be inserted into the subroutines. The directory item which calls for the subroutine must carry in digits 4, 5, and 6 the number of directory lines bearing parameters. The label line of the subroutine should have an indication in the same digits specifying that Mod I or II is to be used. The lines of parameters after each directory item calling for the routine are numbered parameters 1, 2, etc.

C. Keys

The keys indicating which lines of a subroutine are to be modified are stored in the lines following the printer breakpoints sentinel at the end of the subroutine. There is a 6-character key for every modification of the subroutine. The keys are terminated by a half word of Z's (ZZZZZZ) on either the right or left.

The keys specify which line of the routine is to be affected, whether the left-hand side or right-hand side or both are to be altered, and which parameter from the directory is to be used. Keys for Mod I and Mod II are described specifically below.

Keys may extend into the following block of the library if necessary. There is a 60 line (120 key) limit. The lines referred to in the key should be in ascending order of blocks.

Mod I

Mod I adds half or whole words from the directory into the lines of the subroutine. The key for Mod I is

```
KKK L DD
    R
    W          where
```

KKK The line of the subroutine which needs modification

L Left-half of KKK is to be changed

R Right-half of KKK is to be changed

W Whole word KKK is to be changed

DD The number of the parameter in the directory after the directory item specifying the subroutine.

Mod II

Mod II will extract a whole or a half line from the directory into the lines of the subroutine. The key for Mod II is

```
KKK L ED
    R
    W
```

KKK Same as for Mod I

L " " " "

R " " " "

W " " " "

E (extraction) Mod II will be used to modify the lines

D The same as DD for Mod I but this is limited to only 9 parameters.

Note 1. Whenever a subroutine which is to be processed under Mod I or II contains more than one block a blank tape is required on servo 4 during compilation. The compiler does Mod I and II and then proceeds to compile in the normal manner.

Note 2. The label line of a subroutine cannot be modified.

Section 5. Parallel Edit

If there is a servo number in digit 3 of the Omnifax Compiler control word, then information to be edited will be written on this servo during compilation. For every block of compiled output 4 blocks of parallel edit information is put on the tape. This information can later be processed by the parallel edit. The result is an edit, with a heading, date, and page number and relative addresses and coding next to absolute addresses and coding. The programmer will be able to see what he put into the compiler and what the compiler gave him as a result. The Parallel Edit is described on page 32 of the Omnifax manual NYO-6485.

Note: If Mod I or II is used the edit of the relative coding shows the modification as already having been done.

Section 6. Method of Operation

A. Control Word

The Omnifax control word for the compiler is,

```
CXP I 0 0 000 000
  0 0   D
        N
```

D_1 - D_2	CX	Indicating that the Compiler part of Omnifax is to be used
D_3	P	Servo number if output for parallel edit desired
	0	Zero if no parallel edit needed

D_4	I	Servo number if either directory on tape or second special library tape needed.
	0	Zero if neither directory on tape nor special library tape needed
D_5	0	Output servo number
D_6	D	Directory on servo I (D4)
	0	Directory to be typed in on S. C.
	N	Directory is already in the memory, start compiling.
D_7- D_{12}	000000	Not used

B. Servos

Fixed servos:

Servo
3. Library of subroutines (unique to problem)
4. Blank - Temporary storage if any of the subroutines using Mod I or II is longer than one block
6. Library of subroutines (General Library Routines)

Variable servos (any):

a) Servo for Omnifax

b) Servo for parallel edit (if desired). Specified by programmer in digit 3 of the control word

c) Servo for directory and library of subroutines (unique to problem). Specified by programmer in digit 4 of the control word (if desired).

d) Servo for Output. Specified by programmer in digit 5 of the control word.

C. Breakpoints

Breakpoint 5. Force transfer to change a directory item before it is processed.

Breakpoint 7.
- a. Depress breakpoint 7 after a forced transfer on breakpoint 5 if successive words of the directory are to be corrected.
- b. Compiler will keep asking for type-ins.
- c. A return to compiling occurs if
  - 1) REPLACEMENTS is typed in, or
  - 2) ZZZZZZZZZZZZ, indicating the end of the directory is typed in.
- d. Before the machine returns to compiling it will stop on a 90 order.
  - 1) Hit start bar to continue compiling.
  - 2) Clear "C" to restart.

D. Type-Outs

Normally the typewriter is set on normal and the type-outs of significance are directory items. Each item is typed out just before breakpoint 5.

Error Type-Outs

1. REPLACE SKIP occurs if a line of the directory contains 12 zeros. Compiler calls for a type-in to replace this word. If a skip is desired type in 000 S00 000 000.

2. TOO MUCH DIR occurs when the ZZZZZZZZZZZZ sentinel of the directory is not found within 3 blocks. The compiler

will restart automatically and call for a new control word.

3. NOT IN LIB [Directory Item] NEW DIR WD types out if the subroutine called for could not be found on the tape specified. After the type-outs a new directory word type-in is called for.

After the type-in the machine will stop on a 90 order. Hitting the start bar will cause the compiler to process this new word instead of the old one and continue on. Clearing "C" will permit restarting.

4. NOT IN DIR [Directory Item] [External reference or mixed reference line] occurs if some external reference cannot be found in the directory. The machine will stop on a 90 order. If the start bar is hit a NEW LIB WD (New library word) is called for to replace the other external reference and be processed. Clearing "C" will restart.

5. NO BREAKPOINTS [Directory Item] occurs if the compiler cannot find the line of breakpoints sentinel at the end of the subroutine. The compiler will restart. A new control word is called for.

6. CHECK NUMBER [Directory Item] [Contents of Absolute Counter] occurs if the least significant digit of the starting line number in the directory item is not equal to the least significant digit of a special counter. This counter is used to keep track of the length of éach routine.

E. Restarts

Clearing "C" at any time in the middle of the compiler will call for a new Omnifax control word. If a new compiler control word is typed in everything is reset and all the servos used are rewound. An N in D_6 of the new control word indicates that the directory has been already read in and compiling may start.

Section 7. Example

Directory

000) A1Z 002 007 000	Routine A1 on Servo 3 (Special Library Tape)	
001) 050 000 050 000	Parameters	Mod I
002) BCC 032 000 000		Mod II
003) 000 FF0 000 000	Partial fill; next routine starts on a zero (Mod 10) line.	
004) EX0 000 104 010	Routine EX on Servo 6 (General Library Tape)	
005) 000 FFF 000 000	Next routine starts a new block	
006) CCI 000 058 240	Constant Routine located on Servo I Special library tape. (The directory could be the first block on this tape).	
007) 000 F02 000 000	Save two spaces (may need 2 more constants).	
008) F2Z 000 100 300	Routine F2 on Servo 3 (Special Library Tape)	
009) ST0 S00 000 600	Working storage. Necessary if ST is referred to relatively by another routine.	
010) ZZZ ZZZ ZZZ ZZZ	Ending sentinel	

Subroutine

```
000)A10 002 007 001̸
001)100 000 300 000
002)300 060 300 240
003)300 300 300 360
004)U00 312 COX ψψψ      ψψψ is equivalent to 000 absolute
005)BEX 005 COG 001
006)600 000 UOR 000
007)B̸B̸B̸ B̸B̸B̸ B̸B̸B̸ B̸B̸B̸
008)001 W01 002 W01
009)003 L01 004 LE2
010)006 L01 ZZZ ZZZ
```

Before Compilation (went through Mod I): After Compilation

```
000) A10 002 007 001̸                    000 A10 000 007
001) 150 000 350 000                     150 000 350 000
002) 350 060 350 240                     350 060 350 240
003) 350 300 300 360                     350 300 300 360
004) BCC 032 COX ψψψ                     B   272 C   000
005) BEX 005 COG 001                     B   015 C   001
006) 650 000 UOR 000                     650 000 U   000
007) B̸B̸B̸ B̸B̸B̸ B̸B̸B̸ B̸B̸B̸
```

Index - NYU Library of Subroutines

ARITHMETIC ROUTINES

Interpretive Floating Decimal Routine P-550106*
(AFO 000 016 001̸)
A routine to interpret successive lines of three-address instructions as composed floating decimal operations.

Composed Floating Decimal P-551108
(CDO 000 060 001̸)
A one-block routine for performing addition, multiplication, division and composition of numbers expressed in composed floating decimal form.

Floating Point Arithmetic P-550511 and P-550511-1
(FPO 000 120 001̸)
A routine similar to the commonly used floating decimal routine, FR, save that is speeded up by eliminating intermediate composition of numbers, and is shorter to the extent that fewer R instructions need be coded.

Floating Decimal Routine P-550404
(FRO 000 120 001̸)
This is the floating point routine which has been in use longest at N.Y.U. FR must be compiled into lines 880 and following.

Complex Number Arithmetic P-551107
(CEO MOD 060 001̸)
Addition, multiplication, and division of complex numbers. Composed floating decimal requiring routine FR.

Complex Number Arithmetic P-551106
(CXO 000 060 001̸)
Addition, multiplication, and division of complex numbers. Fixed point.

DEBUGGING ROUTINES

Generator for Debugging Routines P-560104
(GRO 000 180 001̸)
A routine to generate a requested debugging routine into available blocks of storage in a subject program.

Locator P-560215
(LCO 000 082 001̸)
A short routine to perform certain tape operations with the use of typed-in control words. Special features include library sub-routine locating and initial reads into any place in the memory.

Address Search P-560718
(ASO 000 039 001̸)
A short routine to search a selected section of the memory and to type out on S. C. Typewriter all instructions containing references to particular addresses.

Automatic Dumping Routine P-550510
(DBO MOD 060 001̸)
A routine to perform automatic storage dumps (17 blocks onto tape) or to give automatic register and storage type-outs.

Memory Dump Comparer P-551104
(DCO 000 264 001̸)
A routine to compare sets of memory dumps from tape putting discrepancies either on S. C. typewriter or on tape. Coding is absolute.

Storage Dump Routines P-550829

(DPO 000 026 00i̸)

(DSO 000 008 00i̸)

DP preserves the contents of registers, dumps a selected region of storage onto a selected tape, and restores the registers.

DS dumps 17 blocks of storage onto servo △.

Q, T, U Tracer - S.C. Typewriter Output P-560105

(TRO 000 060 00i̸)

A routine to monitor a subject program, recording the transfer instructions as they are executed.

Q, T, U Tracer - Tape Output P-560106

(TEO 000 227 00i̸)

A routine similar to TR above

Tracer Counter P-560107

(TCO 000 120 00i̸)

A routine to monitor a subject program recording the changes effected on certain given line locations.

Register A Monitor - S.C. Typewriter Output P-560108

(MNO 000 109 00i̸)

A routine to monitor a subject program, recording the changes effected on the A register.

rA Monitor - Tape Output P-560109

(MEO 000 180 00i̸)

A routine similar to the above MN.

Follower I P-540729-2

(FLO MOD 120 00i̸)

The follower is a diagnostic service routine which controls the execution of a subject program and records on tape the consequence of performing each instruction.

DIFFERENTIAL EQUATION ROUTINES

Differential Equation Solution P-550512
(DFO MOD 069 00~~1~~)
Integration of a system of ordinary differential equations using the Runge-Kutta-Gill method. The arithmetic is floating decimal requiring the use of routine FP.

Differential Equation Solution P-540810
(DYO MOD 062 00~~1~~)
Integration of a system of ordinary differential equations using the Runge-Kutta-Gill method. The arithmetic is fixed points.

Differential Equation Solution - Milne's Method P-560402
(DMO MOD 218 00~~1~~)
Integration of a system of first order differential equations using Milne's method with interval change option. Requires the use of routine FP.

EDIT AND OUTPUT ROUTINES

Sign Edit P-540830
(EAO 000 007 00~~1~~)
A routine to prefix a positive number with a plus sign in place of the Univac zero.

Edit Floating Decimal Composed Number P-550104
(EBO 000 034 00~~1~~)
A routine to edit a composed floating point number into two words -- a signed mantissa with decimal point and a signed base 10 exponent.

Edit Floating Decimal Decomposed Number P-550110
(EPO 000 035 00~~1~~)
A routine to edit a decomposed floating point

number into a signed mantissa with decimal point and a signed base 10 exponent.

Edit Fixed Point Number P-560425

(ESO 000 054 001̸)

A routine to perform round-off, initial zero suppression and arbitrary decimal point location.

Carriage Return Counter P-560719

(R̸CO MOD 026 001̸)

A routine to edit a heading, insert a carriage return upon subsequent entrances until a specified number of carriage returns per page has been reached with a special entrance for filling the remainder of a page with carriage returns.

Packing Routine P-560510

(PAO MOD 060 001̸)

An output routine for an edit routine which will pack a variable number of digits from rA to an output block. Three times as fast as the N.Y.U. Ignore Squeeze routine.

Block Output P-550109

(BLO MOD 010 001̸)

On each entrance to this routine the quantity in rA is stored at the next available location in a specified output block. When the output block is filled it is written on a designated tape.

Block Output [V-Register] P-550108

(BVO MOD 010 001̸)

A routine similar to BL except that the block is filled by pairs of quantities from rV.

EXPONENTIALS AND ROOTS

A^x (Fixed Point) P-550414

(AXO 000 032 001)

A routine to evaluate A^x for

$0 \le A \le 1.$
$0 \le x \le 1.$

10^x P-550416

(LHO 000 040 001)

A routine to evaluate 10^x for $|x| < 50$ with error ± .00008 10^k (k = fractional part of $|x|$). Composed floating decimal (FR).

10^x (Fixed Point) P-560622

(LSO 000 024 001)

A routine to evaluate 10^x for $-13 < x \le 0$. Seven significant figures.

[A University of California, Radiation Laboratory routine incorporated into N.Y.U. Library with permission of the authors.]

10^{-x} FAST (Fixed Point) P-560307

(LVO 000 030 001)

For $0 \le$ fractional part of $x \le 1$ the error is$\sim 5 \times 10^{-5}$.

e^x P-540621

(LPO 000 052 001)

Decomposed floating decimal (FR)

e^x P-560907

(LQO 000 055 001)

Transcribed from LP for use with routine FP.

e^x (Fixed Point [x non-positive]) P-540622

(LTO 000 033 001)

Square Root P-540605
(NAO 000 024 00l̸)
Composed floating decimal (FR)

Square Root P-540603
(NJO 000 023 00l̸)
Decomposed floating decimal (FR)

Square Root P-560912
(NKO 000 024 00l̸)
Transcribed from NJ for use with routine FP.

Square Root (Fixed Point) P-540601
(NSO 000 012 00l̸)

Integral Root : $A^{1/p}$ (Fixed Point) P-550426
(NXO 000 034 00l̸)
$|A| < 1$; $p \geq 2$
(Newton-Raphson iterative procedure.)

IGNORE SQUEEZE ROUTINES

Ignore Squeeze Block Output I P-550329
(GNO MOD 057 00l̸)
On each entrance to this routine the ignores are removed from the word in rA and the result is stored into an output block. (Faster than routine IG.)

Ignore Squeeze Block Output II P-551105
(IGO MOD 034 00l̸)
Same as routine GN in purpose but shorter in space.

INTEGRATION ROUTINES

Integration: Simpson's Rule P-540607
(QNO 000 050 00l̸)

A routine to evaluate $\int_a^b f(x)dx$ given a,b, the interval of integration and a routine for f(x). Decomposed floating decimal - FR.

Integration: Gauss' Method P-550506

(GMO 000 035 001)

Evaluation of an integral using Gauss' method (integrand evaluated at unequal intervals) of order $n(4 \leq n \leq 10)$. Decomposed floating decimal - FR.

Integration: Gauss' Method (Fixed Point) P-550505

(GSO 000 025 001)

Fixed point version of routine GM. G4.... through G- are the routines containing the coefficients for GM and GS.

INTERPOLATION ROUTINES

Aitkin Interpolation P-540811

(IPO 000 062 001)

A routine to perform interpolation of order n using Aitkin's method (restricted to equal intervals). Decomposed floating decimal - FR.

Lagrange Interpolation P-540624

(ILO MOD 069 001)

A routine to perform interpolation of order n using Lagrange's formula (unequal intervals). Decomposed floating decimal - FR.

LOGARITHM ROUTINES

Natural Logarithm P-550830

(LFO 000 044 001)

Composed floating decimal - FR.

Natural Logarithm P-560911
(LGO 000 042 001̸)
Transcribed from LF for use with routine FP.

Natural Logarithm (Fixed Point) P-560326
(LMO 000 029 001̸)
$\frac{1}{100} \ell n\ x$ given with an error $< 7 \times 10^{-5}$.

Logarithm to Base 10 P-550415
(LNO 000 022 001̸)
A routine to find $\log_{10} x$ with $|\text{error}| < .0007$.
Floating decimal - FR.

Logarithm to Base 10 - FAST (Fixed Point) P-560306
(LUO 000 024 001̸)
A routine to find $- \log_{10} x$ for $0 < x < 1$.

Logarithm to Base 2 P-540702
(LJO 000 040 001̸)
Composed floating decimal - FR.

Logarithm to Base 2 (Fixed Point) P-540712
(LXO 000 043 001̸)

MATRICES AND LINEAR EQUATIONS

Most of the matrix codes developed at N.Y.U. are on a separate tape described in report NYO-6484.

Multiplication of Matrices P-560327
(MCO MOD 055 001̸)
Floating point multiplication of two rectangular matrices stored internally.

NUMBER THEORY

Factorization P-560409
(FZO 000 160 001̸)

A routine to find the prime factors of x for $0 < x < 10^{11}$.

POLYNOMIAL ROUTINES

Polynomial Evaluation (Fixed Point) P-540606

(PSO 000 018 001̸)

A routine to evaluate $\sum_{j=0}^{n} a_j x^j$ given n, the a_j's and x.

SORTING ROUTINES

Sorting Routine I P-550827

(STO MOD 032 001̸)

A routine to sort Univac words (numbers and/or letters) into a sequence of increasing magnitude. Up to 968 words may be sorted.

TRIGONOMETRIC ROUTINES

Sine-Cosine P-550107

(TDO 000 052 001̸)

Composed floating decimal - FR.

Sine-Cosine P-550105

(TKO 000 054 001̸)

Decomposed floating decimal - FR.

Sine-Cosine P-560913

(TLO 000 053 001̸)

Transcribed from TK for use with FP.

Sine-Cosine (Fixed Point) P-550413

(TVO 000 041 001̸)

Cosine πx - FAST (Fixed Point) P-560308

(TUO 000 020 001̸)

For $0 \leq x < 1$, the error is $\sim 1 \times 10^{-4}$.

<u>Tan^{-1}x (Fixed Point)</u> P-550902

(TYO 000 057 00~~1~~)

Continued fraction approximations, error $< 10^{-10}$.

<u>Tan^{-1}x (Fixed Point)</u> P-550901

(TZO 000 025 00~~1~~)

Polynomial approximation,error $\sim 10^{-8}$. (Faster but less accurate than TY.)

* N.Y.U. File Number

c.3

NYU NYO-7692
Ragusa
The NYU Omnifax compiler
and library of subroutines.